

Facilitating Multimedia Database Exploration through Visual Interfaces and Perpetual Query Reformulations

Wen-Syan Li K. Selçuk Candan[†] Kyoji Hirata Yoshinori Hara

C&C Research Laboratories, NEC USA, Inc.
110 Rio Robles, San Jose, CA 95134, USA
Email: {wen,candan,hirata,hara}@ccrl.sj.nec.com

Abstract

Since multimedia retrieval is based on similarity calculations of semantics and media-based search, exact matches are not expected. We view querying multimedia database as a combination of IR, image matching, and traditional database query processing and it should be conducted in a way of perpetual query reformulation for honing target results. In this paper we present a hybrid multimedia database system, which employs a hierarchical database statistics structure for both query optimization and reformulation analysis without adding additional query processing cost. The contribution of this work is the integration of database methodology, user interface design, IR concepts, and image matching techniques for facilitating multimedia database exploration.

1 Introduction

Query processing in multimedia databases is different from that in traditional database systems. Because contents stored in traditional database systems are rather precise, query results are certain. In information retrieval, documents are represented as keyword lists. To retrieve a document, systems compare keywords specified by users with the documents' keyword

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the VLDB copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Very Large Data Base Endowment. To copy otherwise, or to republish, requires a fee and/or special permission from the Endowment.

Proceedings of the 23rd VLDB Conference
Athens, Greece, 1997

lists. Images, in a similar manner, are represented as features. Image matching is carried out through comparing these features. In both information retrieval and image matching, results are based on similarity calculations; comparing similarity between semantic meanings of keywords and image features respectively.

An image can be viewed as a compound object with visual features, structural layout (spatial relationships), and semantics. Because query processing in multimedia databases is based on similarity calculations on all these aspects, it can be viewed as an integration of information retrieval notions described in [1] (exploratory querying, inexact match, query refinement) with ORDBMS or OODBMS notions (recognition of specific concepts, variety of data types, spatial relationships between objects). We argue that querying multimedia databases should be conducted in an "exploratory" fashion and the system needs to provide useful feedback to facilitate query reformulations because exact matches are *not* expected. We next present our problem statement and give an overview of our solutions to it.

1.1 Problem Statement

We see that two important functionalities for multimedia database systems need to be enhanced:

Query interfaces that bridge the gap between users' target results and complex query languages and multimedia modeling schemes: There are two major development directions in query specification mechanisms for multimedia databases. One direction is manipulation through SQL-like query languages, such as Multimedia Query Language[2], which are precise to computers but too complicated for users to use directly and these languages do not visualize target results. Another direction is specifying queries using more natural specification methods, such as query by image examples or using natural languages. Query by image ex-

[†]This work was performed when the author visited NEC.

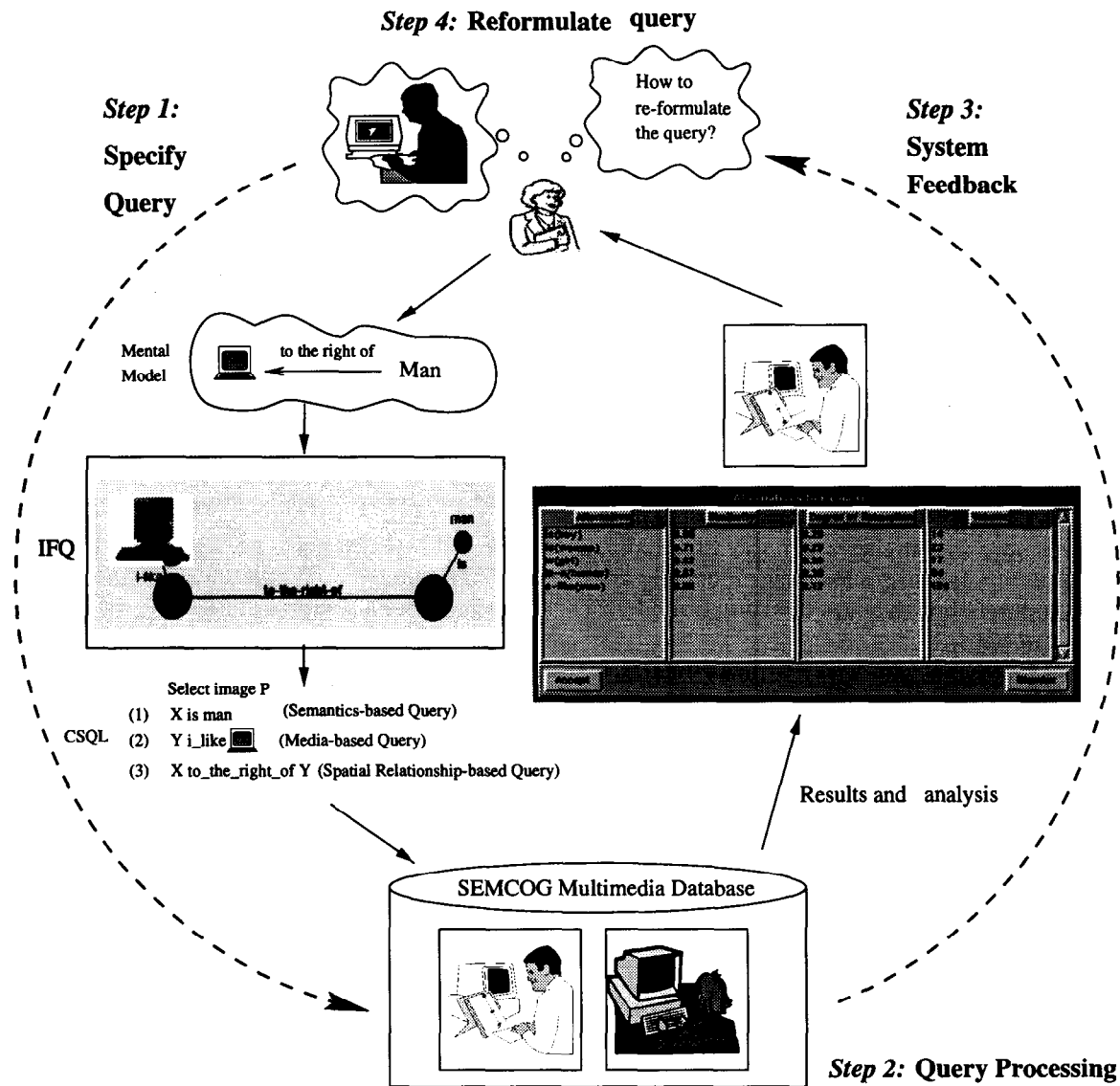


Figure 1: System Facilitated Multimedia Database Exploration Cycle in SEMCOG

ample mechanism is to support “drawing board”-like interfaces for users to provide image examples or drawing. This method can visualize users’ target images, but it has the drawback of low precision since users’ drawings are usually ambiguous to image matching engines. Another drawback of query by image example approach is that it cannot specify *concepts*, such as transportation and appliances. Another more “natural” way for image retrieval is describing target image semantics using somewhat “natural languages.” The approach is usually too tedious so that usually only trained domain experts are capable of using this technique. It is essential to facilitate the interactions between human and computers in querying multimedia databases.

System feedback for facilitating query reformulations to bridge the gap between data contents

and users’ target results: Based on similarity calculation, multimedia retrieval should be conducted in the form of “exploration”, similar to notions of IR. System feedback is very important for users to reformulate their initial queries. Most existing systems only provide a list of images ranked by similarity. The information on how to reformulate queries is not provided and query reformulations are performed by way of “trial-and-error”. It is important to apply database techniques to not only query processing and optimization but also system feedback.

1.2 Our Approach

Our multimedia database query model, *system facilitated exploration*, consists of four steps: specify query, query processing, system feedback, and query reformulation, as illustrated in Figure 1. These four steps form

a cycle of perpetual query reformulation for honing target images. In this cycle, a visual query interface is employed to assist users in specifying queries without being required to be aware of underlying complex query languages and multimedia modeling scheme. If there is no exact match, there exists a gap between the users' queries and the actual contents stored in the database. System feedback is then presented to the users to facilitate query reformulations. Through the cycle of perpetual query reformulations, the users explore multimedia databases for honing target images.

Figure 1 shows an example in which a user wants to retrieve an image containing a man and a computer and the man is to the right of the computer. There is a gap between the user's mental model and the actual image modeling in the database system. User's mental model is more natural (in terms of perception), but it is rather ambiguous to computers. On the other hand, direct manipulation languages for image modeling are precise to computers but they are too complicated for users to use directly.

IFQ (In Frame Query)[3], a visual query interface used in our system, is an attempt to bridge this gap. With IFQ, users can specify their mental models using both visual examples or semantics as shown in the middle of Figure 1. The user specifies the query in the way of interacting with IFQ while the corresponding query is automatically generated so that users are not required to be aware of complex multimedia schema design and query language syntax. IFQ also visualizes target images, unlike other graphical query interfaces.

In most cases, there may not be exact matches, but there exists other images which partially match users' specifications. Our system provides useful feedback to assist users in reformulating queries. The system feedback includes (1) query reformulation alternatives for all three types of query conditions; (2) how *discriminating* each condition is; and (3) expectation of finding target images with such reformulation. For the example in Figure 1, the system shows the query reformulation alternatives for *is(man)*, such as *is(woman)*, *is-a(human)*, and *is(child)*; and numbers of images containing such objects. With this analysis, users can better visualize image content distributions related to the initial queries. In our system, query reformulations are perpetual and facilitated through system feedback.

The rest of this paper is organized as follows: We first present the architecture, query language and interface of our system. We also show the design of database statistics storage and indexes for query optimization and reformulation facilitation. In Section 3, we explain how our system provides useful feedback to assist users in honing target images through analyzing and presenting query reformulation alternatives. In Section 4, we discuss existing systems and compare

them with ours. Finally we offer our conclusions.

2 SEMCOG Image Database System

SEMCOG (SEMantics and COGnition-based image retrieval) is an image database system developed at NEC C&C Research Laboratories. In SEMCOG, image matching can be performed at both the whole image and object levels and it integrates semantics and image-based approaches to support a higher flexibility in query specifications. For example, a query "Retrieve all images in which there is a man to the right of a car and the man looks like this image" can be posed. In this section, we present the design and functionalities of SEMCOG.

2.1 Architecture and Components

Figure 2 shows the operation and data flows in SEMCOG. Three layers of functions are provided. The client functions are supported by IFQ visual query interface, including query specification, result display, and query generating. The server functions are supported by five modules to augment the underlying database system multimedia manipulation and search capability. SEMCOG also maintains database statistics for query optimization and query reformulation facilitation. The components of SEMCOG and their functionalities are as follows:

Image Data Query Processor is used to perform image matching task. COIR (Content-Oriented Image Retrieval)[4] is the image processing module employed in SEMCOG. It performs image matching at the object level by identifying regions defined as homogeneous (in terms of colors) and continuous segments identified in an image. The main task of COIR is to identify image regions. To identify objects within an image and their semantics, COIR needs to consult *Image Component Catalog* to match with existing image object samples to determine image semantics.

When an image is *registered* at SEMCOG, some image descriptions, such as the image size, image format, registration date, are automatically extracted, while some metadata, such as semantics of images, cannot be extracted automatically. *Image Semantics Editor* interacts with COIR to specify the semantics of an image. The steps involved in image semantics extraction and editing are as follows: (1) COIR identifies image regions of an image. (2) COIR recommends candidate components and their corresponding semantics by consulting *Image Component Catalog*. (3) The user confirms or modifies the results by COIR. (4) The user may select objects to store in the image component catalog as "representative samples" for future use. *Image Component Catalog* is built incrementally in this manner. (5) COIR identifies spatial relations of the

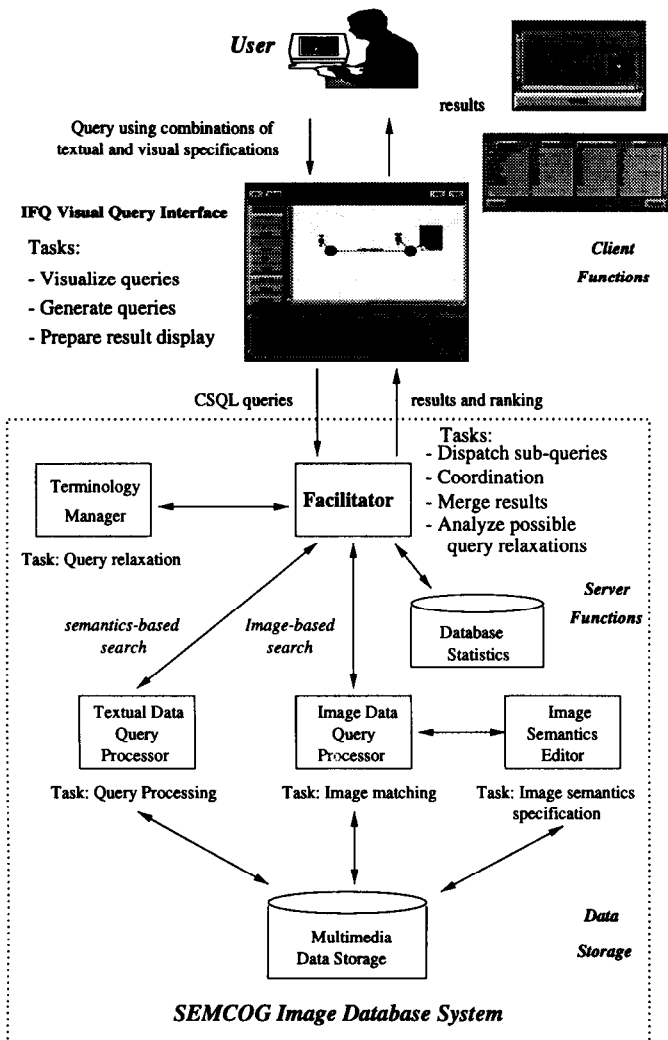


Figure 2: System Architecture of SEMCOG

objects based on the comparison of objects' four outermost points or weight centers.

Terminology Manager maintains a terminology base that is used for query term relaxation. Consulting with *Terminology Manager*, SEMCOG can find synonyms for a given term and access IS_A hierarchies. For predicates, such as “*is_a transportation*” or “*s_like man*”, the *Terminology Manager* returns multiple resulting terms with their ranking. Note that we do not need to build a terminology database, many existing lexical reference systems, such as WordNet[5], can be employed.

Textual Data Query Processor processes queries concerning image or object semantics, which are “non-image-matching operations”. All the query statements except those with *i_like* (image like) and *contains* are semantics-based. The query processing performed here is the same as that in traditional relational DBMSs, except the additional task associated with query relaxation predicates, such as *is_a*.

We intend to design SEMCOG as a hybrid architecture so that many modules can be plugged in as components to handle multiple media. *Facilitator* coordinates the interactions between components. It performs the following tasks: (1) Query relaxation: It reformulates query statements containing *is_a* or *s_like* predicates by consulting the *terminology manager*; (2) Query forwarding: It forwards query statements containing *i_like* or *contains* (visually) predicates to the *Image Data Query Processor* to perform image matching while forwarding other non-image-matching operation statements to *Textual Data Query Processor*; and (3) Result integration: It merges results of image-matching query statements and non-image-matching operation statements. This task includes computing overall similarity values and eliminating images based on specified filtering constraints.

2.2 Query Language

CSQL (Cognition and Semantics Query Language) is the underlying query language used in SEMCOG. CSQL is similar to SQL, but contains additional predicates for image matching and semantics-based query conditions with various levels of strictness. CSQL provides a greater flexibility for multimedia query specifications. The image retrieval related predicates in CSQL include: (1) *Semantics-based predicates*: *is* (e.g. man vs. man), *is_a* (e.g. car vs. transportation), and *s_like* for “semantics like” (e.g. car vs. truck); (2) *Media-based predicates*: *i_like* for “image like” that compares visual signatures of two arguments and *contains*; and (3) *spatial relationship-based predicates*: *above*, *below*, and etc.

2.3 Visual Query Interface

Much work in multimedia databases has focused on developing more powerful languages or modeling scheme. In contrary, research in query interfaces has not been equally promoted. As a result, multimedia database applications can not fully utilize these advanced languages and modeling schemes since they are too complicated for users to use directly or to understand.

IFQ is a query generator and a visual query interface, rather than a “graphical” query interface used by most existing systems. It allows users to pose queries using icons and menus in a drag-and-drop fashion. It also visualizes target images and generates corresponding complex queries as specification process progresses. Figure 3 shows a query “Retrieve all images in which there is a man to the right of a transportation and the transportation looks like this image” posed using IFQ. Note that the corresponding query consists of three types of query condition, semantics-based, cognition-based, and scene-based queries.

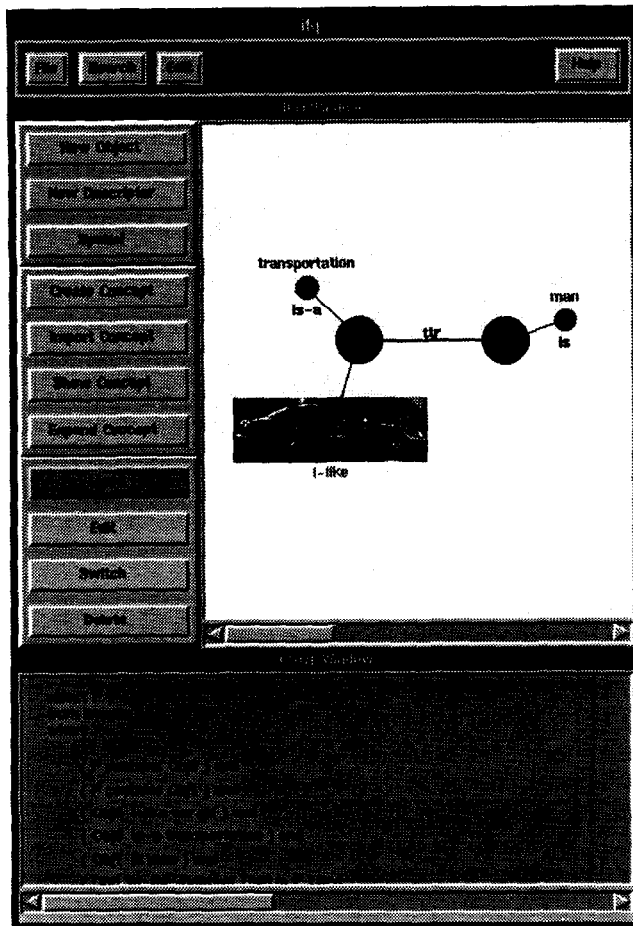


Figure 3: IFQ Query Interface

The query specification process in IFQ consists of three steps: introducing image objects, describing them, and specifying their spatial relationships. In IFQ, objects are represented as bullets and descriptors are represented as small bullets to attach to the objects to describe their visual and semantics properties. The IFQ query in Figure 3 is posed as follows: The user introduces the first object in the image and then further describes the object by attaching “i-like *image*” and “ISA *transportation*” descriptors. After the user specifies a image path or provides a drawing, the interface automatically replaces the descriptor with the thumbnail size image the user specifies, a car in this example. Next, the user introduces another object and describes it using the “is *man*” descriptor. Finally, the user describes the spatial relationship between these two objects by drawing a line, labeled by *to-the-right-of*, from the man object to the transportation object. While the user is specifying the query using IFQ, the corresponding CSQL query is automatically generated in the CSQL window. With IFQ, users pose queries by simply clicking buttons, dragging and dropping icons, and filling in pop-up menus.

A study by H. Nishiyama et. al[6] pointed out that

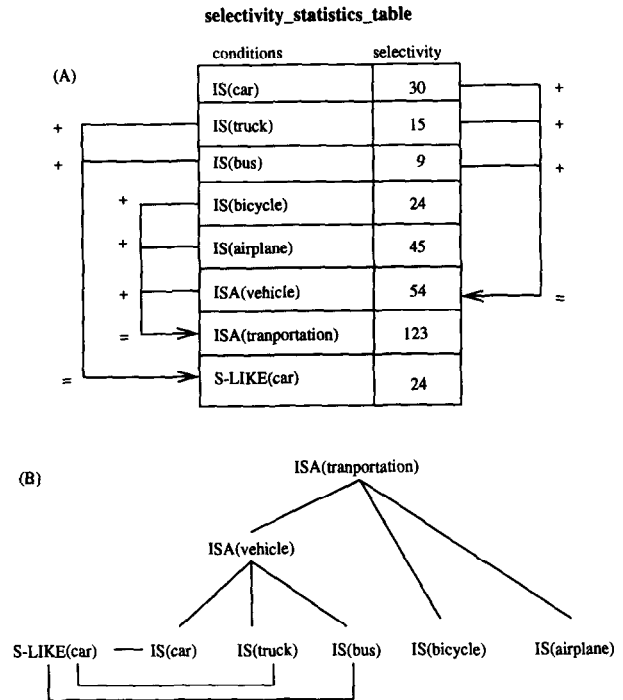


Figure 4: Selectivity as a Hierarchical Structure

two patterns of end-users in their visual memory when they view paintings or images. The first pattern consists of roughly the whole image, whereas the second pattern concentrates on specific objects within the image, such as a man or a desk. This study supports the development of IFQ, which matches users’ mental models with image representation models in multimedia databases.

2.4 Database Statistics

Like most query optimizers in traditional database systems, *Facilitator* also periodically collects database statistics to improve query processing performance; especially using selectivity for query statement execution sequence arrangement. One unique function provided by *Facilitator* is to recommend query reformulation alternatives and analyze possible outcomes so that users can visualize distributions of related images. In order to provide such functionalities, *Facilitator* needs to maintain selectivity statistics on object semantics.

In Figure 4, we show a selectivity hierarchical structure for objects related to transportation. The hierarchical structure is built based on the semantics in *Image Semantics Database* and knowledge in *Terminology Base*. There are two types mechanisms employed to generate *selectivity*, depending on frequency of update and insertion. For the semantic terms which are less frequently updated and inserted, selectivity statistics are gathered periodically, rather than in real time. For those semantic terms which are frequently updated and inserted, selectivity statistics are gathered daily or

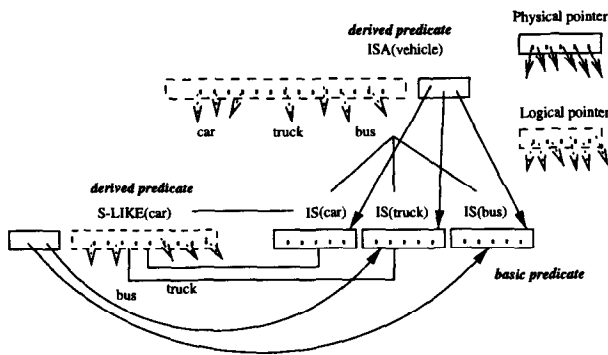


Figure 5: Hierarchical Structured Logical Pointers

in real time through view management and triggers. More detail on how to use the database statistics is discussed later in Section 3.

2.5 Constructing Indexes

Since multimedia database exploration is based on computer human interactions, response time is very important. In order to improve response time, two techniques have been applied: “hierarchically structured indexing” and “role-based query processing and reformulation”. These two techniques are designed to take advantages of the fact that related predicates are “clustered” as a hierarchical structure. As shown in Figure 4, there are 8 predicates related to the “transportation” concept. Two useful properties with this clustering are:

1. Some predicates can be translated into a disjunction form of other predicates. For example, $ISA(vehicle)$ can be translated into a disjunction form of $IS(car)$, $IS(truck)$, and $IS(bus)$.
2. These clustered predicates can be “reformulated” from/to each other. For example, $IS(car)$ can be reformulated horizontally to $S-LIKE(car)$ or vertically to $ISA(vehicle)$ or $ISA(transportation)$.

“Hierarchically structured indexing” technique is to take advantage of the first property. In order to provide quick response time, we build indices for each predicate. There are two types of predicates: *basic* predicates and *derived* predicates. All $IS()$ predicates are *basic* predicates, while $S-LIKE()$ and $ISA()$ predicates are *derived* predicates. As shown in Figure 5, $ISA(vehicle)$ can be derived from a disjunction of $IS(car)$, $IS(truck)$, and $IS(bus)$ and $ISA(transportation)$ can be derived from a disjunction of $ISA(vehicle)$, $IS(bicycle)$, and $IS(airplane)$.

We first build an index for *basic* predicates on object semantics (e.g. car, bicycle, airplane). We call this type of indices as “physical” or “direct” indices. Indexes for $IS(car)$, $IS(car)$, and $IS(truck)$ are “physical” pointers pointing to objects which satisfy $IS(car)$,

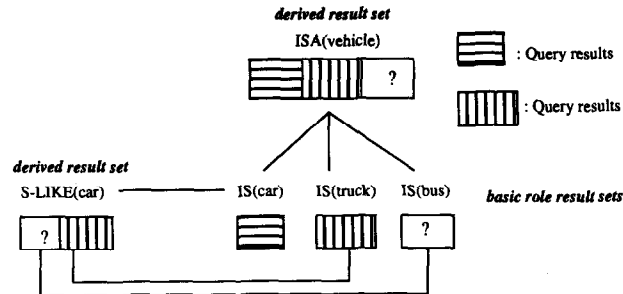


Figure 6: Role-based Query Processing

$IS(car)$, and $IS(truck)$. The indices for *derived* predicates, which are built using “physical” indices, are “logical” indices. As shown in Figure 5, $ISA(vehicle)$ logically has indexes for tuples whose semantics are car, truck, or bus. However, the physical index consists of only three pointers pointing to the physical indices of $IS(car)$, $IS(car)$, and $IS(truck)$. Similarly, $S-LIKE(car)$ consists of only two physical pointers while logically consists of pointers to all objects whose semantics are truck or bus.

Another technique we have applied for query optimization is role-based query processing. This technique is an attempt to take advantage of the second property: one predicate can be “reformulated” from/to each other. Similar to the concept described in [7], we categorize query results into *basic role result sets* and *derived result sets*. *Basic role result sets* include all results produced by IS predicates, such as $IS(car)$, $IS(bus)$, and $IS(truck)$. The basic role result sets are atomic. On the other hand, the result for $S-LIKE(car)$ can be derived as the summation of $IS(truck)$ and $IS(bus)$. Similarly, the result for $ISA(vehicle)$ can be derived as the summation of $IS(car)$, $IS(truck)$, and $IS(bus)$. This is illustrated in Figure 6.

With this property, many results in the cache can be utilized to construct new results for reformulated query. For example, a user submits a query with a condition $IS(car)$ and then reformulates it to $IS(truck)$. If the user next relaxes the condition to $ISA(vehicle)$, the system only needs to compute the result for $IS(bus)$ using pre-built pointers and combine it with the role result sets of $IS(car)$ and $IS(truck)$.

3 Multimedia Database Exploration

In this section, we present a novel approach to exploring multimedia databases with system facilitation; that is beyond browsing, the functionality supported by most existing systems.

3.1 Query Reformulations as Navigations

In this paper, we use “query reformulation” rather than the terms “query relaxation” (used by the

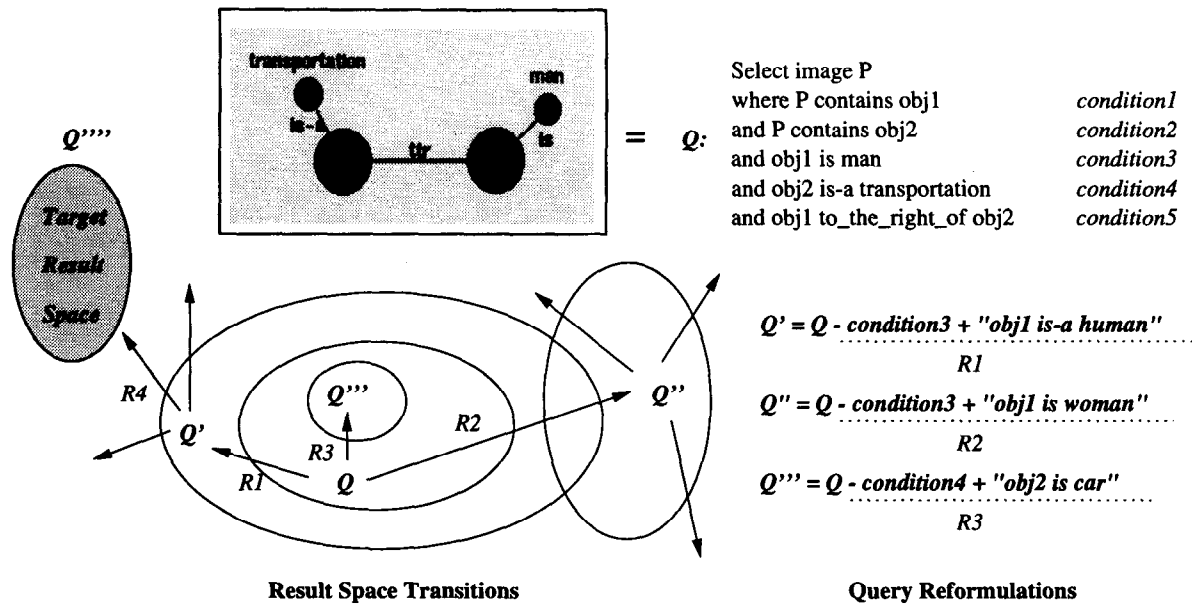


Figure 7: State Transitions of Query Reformulations

database community) or “query refinement” (used by the IR community) because users can either relax (e.g. *IS(car)* to *ISA(transportation)*), tighten (e.g. *ISA(transportation)* to *IS(car)*), or change query conditions (e.g. *IS(car)* to *IS(boat)*). A query may contain multiple conditions and there exists result space corresponds to each query.

We view multimedia database exploration as transitions of corresponding query result space. By reformulating a query, the result space that a user sees is shifted from one to another accordingly. For example, the query *Q* in Figure 7 contains five conditions; each one has many reformulation alternatives. We show three possible reformulation examples, *R1*, *R2*, and *R3*, and their results. Each result space transition corresponds to a query reformulation. Through perpetual query reformulations, the user *navigates* through the possible result space to explore multimedia databases.

Most existing systems provide multimedia database exploration through browsing. We argue that browsing for multimedia database exploration cannot scale up when the number of images is huge. Many systems support so-called user feedback functionalities to allow users to select one of candidate images as new image example for new search. We see that system facilitation should include presenting reformulation alternatives, visualizing image content distributions, and providing estimated outcome. With this information, users are facilitated to “navigate” from current result space toward the target result space, rather than ad-hoc trying and error. As shown in 7, the user first applies reformulation alternative *R1*, and then applies *R4* to reach his/her target result space.

3.2 Reformulation Analysis

Since there are three types of predicates; namely, semantics-based statements, image-matching-based statements, and spatial relationship-based statements, three types of reformulation analysis are supported. We next give details on how we conduct reformulation analysis for each type of condition.

3.2.1 Semantics-based Statement

Figure 8 shows an analysis of reformulation for a query which contains two semantics-based statements: *is(man)* and *is(car)*.

The analysis consists of two levels of information. 221.40 and 190.43 represent the high level “degree of expectation” for reformulating *is(car)* and *is(man)* conditions respectively in terms of the possibility of finding target images. The user can then click on the numbers to see detail level of analysis and reformulation alternatives.

As shown in Figure 8, *is(car)* has five reformulation alternatives and each reformulation alternative has a value representing the similarity between each alternative condition and *is(car)*. For example, the similarity between *is(truck)* and *is(car)* is 0.60 and the similarity between *is(bicycle)* and *is(car)* is 0.40. The lower the similarity value is, the higher degree of relaxation is. The similarity can be viewed as *quality of relaxation*.

Each alternative condition also has an entry of *volume*, which represents the *selectivity* of this alternative condition: the number of image objects whose semantics satisfy this condition. This selectivity values are derived and maintained as we described earlier in Section 2.4. This volume values can be viewed as *quantity*

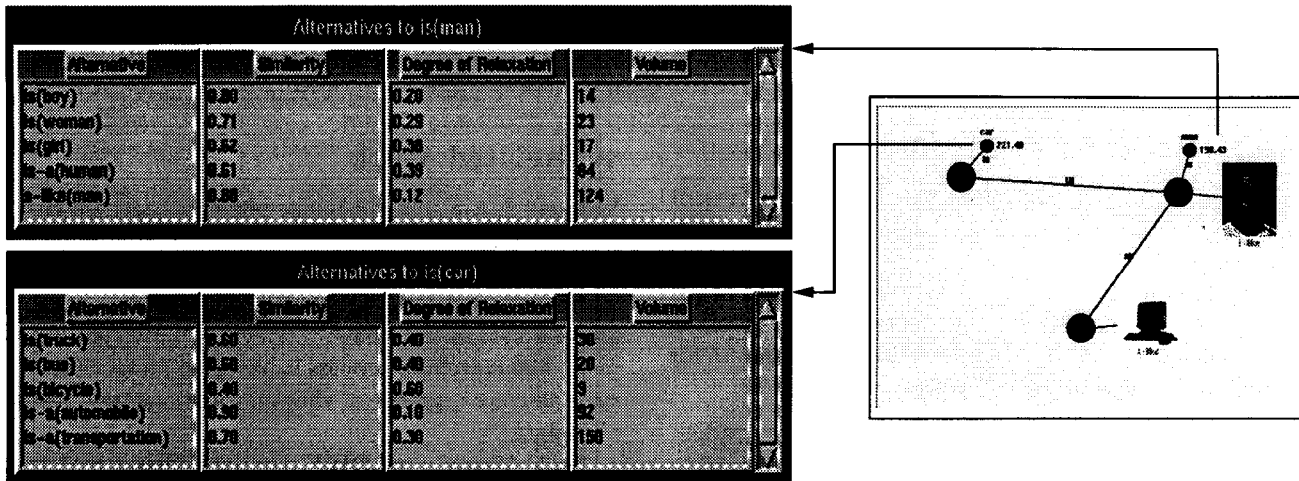


Figure 8: Query Reformulation Alternative Analysis for Semantics-based Statements

of relaxation.

The expectation value is summation of multiplication of similarity value and volume for each alternative condition. An initial query Q consists of n conditions, denoted as C_1 to C_i and for each condition C_i , there exists j reformulation alternatives, denoted as $RC_{i,1}$ to $RC_{i,j}$. The formula for calculating expectation value of reformulating C_i is defined as follows:

$$Expectation_{C_i} = \sum_{j=1}^n (Similarity(C_i, RC_{i,j}) \times Volume(RC_{i,j}))$$

where the term $Similarity(C_i, RC_{i,j})$ is the similarity between conditions C_i and $RC_{i,j}$ and the term $Volume(RC_{i,j})$ is the selectivity of alternative condition $RC_{i,j}$. Note that this formula is similar to the general formula used in IR community, such as the formula used in [8]. The $Volume$ here corresponds to *keyword frequency* in IR and $Similarity$ here corresponds to the similarity between *keywords in document* and *user-specified keywords* in IR.

Many lexical reference systems, such as Wordnet[5], organize words based on their semantic meanings and provide synonyms and homonyms for a given word and their “distance” values. The “distance” values can be used as measures of their similarity, although it is arguable. We denote the distance value of two words X and Y as $Distance(X, Y)$, which can be normalized to a range between 0 and 1. The $Similarity(C_i, RC_{i,j})$ in the above formula is calculated based on these “distance” values as follows:

- Similarity between $IS(X)$ and $IS(Y)$ is $Distance(X, Y)$.
- Similarity between $IS(X)$ and $S-LIKE(X)$ is $\frac{\sum_{i=1}^n Distance(X, S_i)}{n}$ where n is the number of semantically similar terms of X and S_i is the synonyms of X . Examples of this type of reformu-

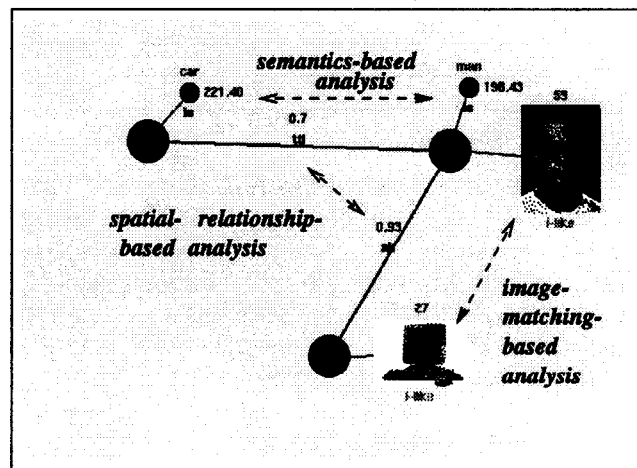


Figure 9: Complete Query Reformulation Analysis

lation include $IS(car)$ and $S-LIKE(car)$ and S_i could be truck, bus, and etc.

- Similarity between $IS(X)$ and $ISA(Y)$, where Y is a hyponym of X , is $\frac{\sum_{i=1}^m Distance(X, S_i)}{m}$ where m is the number of hypernyms of Y excluding X and S_i is the hypernyms of Y excluding X . Examples of this type of reformulation include $IS(car)$ and $IS-A(transportation)$. In this example, Y is transportation, X is car, and S_i could be truck, bus, airplane, and etc.

3.2.2 Image-based Statement

Since image matching is based on similarity calculation, reformulation of image-based statement is by way of adjusting similarity threshold. The system presents feedback as the numbers of image objects satisfying each image-based statement. Figure 9 shows that there are 59 image objects matching with the man’s picture and 27 image objects matching with the computer image, given a image matching threshold. With

this analysis, users can visualize how “discriminating” each image-based statement is and make adjustments on the similarity threshold.

3.2.3 Spatial Relationship-based Statement

For spatial relationship-based statements, we present users with the information on how “discriminating” each spatial relationship constraint is. The degree of ‘discrimination’ is measured by the percentage of tuples that do not satisfy the constraint. For example, an image database contains the following metadata for image object spatial relationships:

Image	Object1	Spatial_relation	Object2
I001	man	to_the_left_of	car
I002	tree	to_the_right_of	car
I003	man	to_the_left_of	airplane
I004	man	to_the_left_of	car
I005	man	to_the_right_of	bicycle
I006	man	to_the_left_of	airplane
I007	man	to_the_left_of	car

For the query shown in Figure 7, there are 6 tuples which satisfy the conditions 3 and 4. However, the condition 5 eliminates 5 of 6 tuples from being considered as an answer. Thus the degree of “discrimination” for the condition 5 is 83% (5/6). Note that we do not need to consider the layout of image I002 since its objects are not man and transportation.

3.3 Design Considerations

One important consideration for implementing such a functionality is not adding much additional query processing costs to the system. For the analysis for image-matching-based and spatial relationship-based statements, the information needed is collected during the query processing time, without requiring additional query processing efforts. For the analysis for semantics-based statements, the information needed is stored in *Database Statistics Storage* described in Section 2.4 and updated through view maintenance and periodical database selectivity statistics collection, rather than exploring all possible query reformulation alternatives in real time, which can be very expensive.

Figure 9 shows a window dump of a complete analysis of reformulation alternatives for all three types of statements. Users have an option to visualize the analysis for all three types of statements (as shown in Figure 9) or only one type of statements, say semantics-based statement (as shown in Figure 8). When users request the system to present complete query reformulation analysis results, the users need to be aware of the fact that the analysis for these three types of statement are actually based on different aspects and

scales. We are still experimenting various ways to visualize analysis of reformulation alternatives for better usability.

4 Related Work

Virage[9] is a system for image retrieval based on visual features, such as color, shape, and texture. Virage has an SQL-like query language and a form-based query interface called VIV. QBIC[10], developed at IBM, is another system for image retrieval using visual examples. The matching process is based on image features, such as colors, textures, shapes, locations, and layout of images.

Both Virage and QBIC support image matching and keyword-based retrieval functionality on the *whole* image level. One major difference between our system and theirs is that SEMCOG supports retrieval based on image matching and semantics at both the whole image and object levels. SEMCOG also allows users to specify more relaxed query conditions, such as *IS.A*.

PESTO[11] is another project at IBM. It focuses on integrating and browsing/querying images from heterogeneous and distributed information sources. PESTO is similar to IFQ to SEMCOG. PESTO allows users to specify tables and attributes for join/project and other aggregation functions for multimedia information retrieval. PESTO is a more complete multimedia database query/browsing interface. On the other hand, IFQ aims at providing SEMCOG computer human interaction functionalities, such as query specification and query reformulation facilitation.

SCORE[12] is a similarity-based image retrieval system developed at University of Illinois at Chicago. This work focuses on the use of a refined ER model to represent image contents and how to calculate similarity between E-R representations of images stored and query specifications. However, SCORE does not support image-matching.

VisualSeek [13] is a content-based image query system developed at Columbia University. VisualSeek uses color distributions to retrieve images. Although VisualSeek is not object-based, it provides region-based image retrieval: users can specify how color regions shall be placed with respect to each other. VisualSeek also provide image comparisons and sketches for image retrieval. VisualSeek is designed for image matching; it does not support image retrieval based on semantics.

Chabot[14] at UC Berkeley is initiated to study storage and retrieval of a vast collection of digitized images from the State of California Department of Water Resources. Chabot provides a form based browser where users can provide metadata, keywords, concepts, and/or color-distributions to retrieve images.

Chabot also supports concept definition functionality. Chabot focuses on whole image matching and query on image metadata.

5 Conclusions

We see that querying multimedia databases is an integration of image matching, IR, and traditional database query processing. We argue that image retrieval should be conducted in the way of system-facilitated exploration through perpetual query reformulation for honing target images. Our multimedia database exploration cycle consists of four steps, *specify query*, *query processing*, *system feedback*, and *query reformulation*. The contributions of this work include (1) Query interface to bridge the gap between users' target results and complex query languages and image modeling scheme; and (2) system feedback for facilitating query reformulations to bridge the gap between data contents and users' target results.

Acknowledgments

We would like to express our sincere appreciation to Peter Scheuermann and Chris Clifton for their invaluable comments and suggestions on this work.

References

- [1] Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Welsey Publishing Company, Inc., 1989.
- [2] S.C. Kau and J. Tseng. MQL - A Query Language for Multimedia Databases. In *Proceedings of 1994 ACM Multimedia Conference*, pages 511-516, 1994.
- [3] Wen-Syan Li, K. Selçuk Candan, K. Hirata, and Yoshi Hara. IFQ: A Visual Query Interface and Query Generator for Object-based Media Retrieval. In *Proceedings of the 1997 IEEE Multimedia Computing and Systems Conference*, pages 353-361, Ottawa, Ontario, Canada, June 1997.
- [4] Kyoji Hirata, Yoshinori Hara, H. Takano, and S. Kawasaki. Content-Oriented Integration in Hypermedia Systems. In *Proceedings of 1996 ACM Conference on Hypertext*, March 1996.
- [5] G. A. Miller. WordNet: A Lexical Databases for English. *Communications of the ACM*, pages 39-41, November 1995.
- [6] Haruhiko Nishiyama, Sumi Kin, Teruo Yokoyama, and Yutaka Matsushita. An Image Retrieval System Considering Subjective Perception. In *Proceedings of the 1994 ACM SIGCHI Conference*, pages 30-36, Boston, MA, April 1994.
- [7] Peter Scheuermann and Eugene I. Chong. Role-based Query Processing in Multidatabase Systems. In *Proceedings of the International Conference on Extending Database Technology*, pages 95-108, March 1994.
- [8] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference*, Denmark, 1992.
- [9] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Jain, and C.-F. Shu. The Virage Image Search Engine: An Open Framework for Image Management. In *Proceedings of the SPIE - The International Society for Optical Engineering: Storage and Retrieval for Still Image and Video Databases IV*, San Jose, CA, USA, February 1996.
- [10] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. Query by Image and Video Content: The QBIC System. *IEEE Computer*, 28(9):23-32, September 1995.
- [11] M. Carey, L. Haas, V. Maganty, and J. Williams. PESTO: An Integrated Query/Browser for Object Databases. In *Proceedings of the 1996 VLDB Conference*, Bombay, India, September 1996.
- [12] Y. Alp Aslandogan, Chuck Thier Clement Yu, Chengwen Liu, and Krishnakumar R. Nair. Design, Implementation and Evaluation of SCORE. In *Proceedings of the 11th International Conference on Data Engineering*, Taipei, Taiwan, March 1995. IEEE.
- [13] John R. Smith and Shin-Fu Chang. VisualSEEK: a Fully Automated Content-based Image Query System. In *Proceedings of the 1996 ACM Multimedia Conference*, pages 87-98, Boston, MA, 1996.
- [14] Virginia E. Ogle and Michael Stonebraker. Chabot: Retrieval from a Relational Database of Images. *IEEE Computer*, 28(9):40-48, September 1995.